Rapid Adjudication of Static Analysis Meta-Alerts During Continuous Integration (CI)

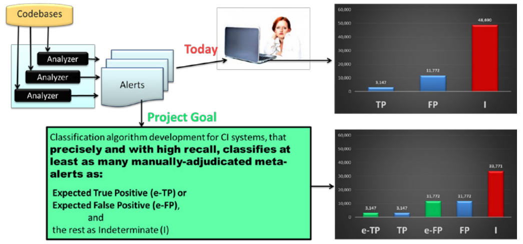
Problem

Manual adjudication of static analysis meta-alerts requires too much effort in short CI build and PR-approval time frames to address many (if any) of them. This problem is technically challenging. Developing a new static analysis to ***precisely*** match flaws in different version of Java or C++ code requires language-specific algorithms, and the matching must be fast to work in a CI/CD system. Also, when cascading is ***imprecise***, mis-labeled data worsens classifier performance, and no effective systems exist that use automated classifiers for multiple static analysis tools in a CI system.

Solution

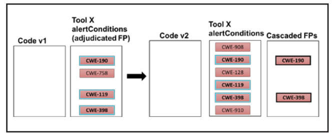
The solution involves (1) a system that supports classification integrated with CI, and builds on the SCAIFE API and implementation we developed for an extensible architecture that supports classification, and (2) precise cascading algorithms for C++ code.

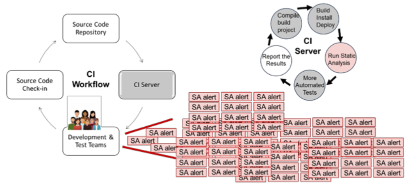
We (1) designed a model for integrated SCAIFE-CI systems, including SCAIFE changes, performance measures, and new classifier features; (2) implemented parts of the design (collaborators tested and reviewed subsequent versions); (3) performed an experiment using diff-based (imprecise) cascading and generated data for comparison to precise cascading. Future plans are to develop a precise cascading algorithm, improve classifiers, and fully integrate them.



Classification algorithm development for CI systems, that **precisely and with high recall, classifies at least as many manually-adjudicated meta-alerts as:**

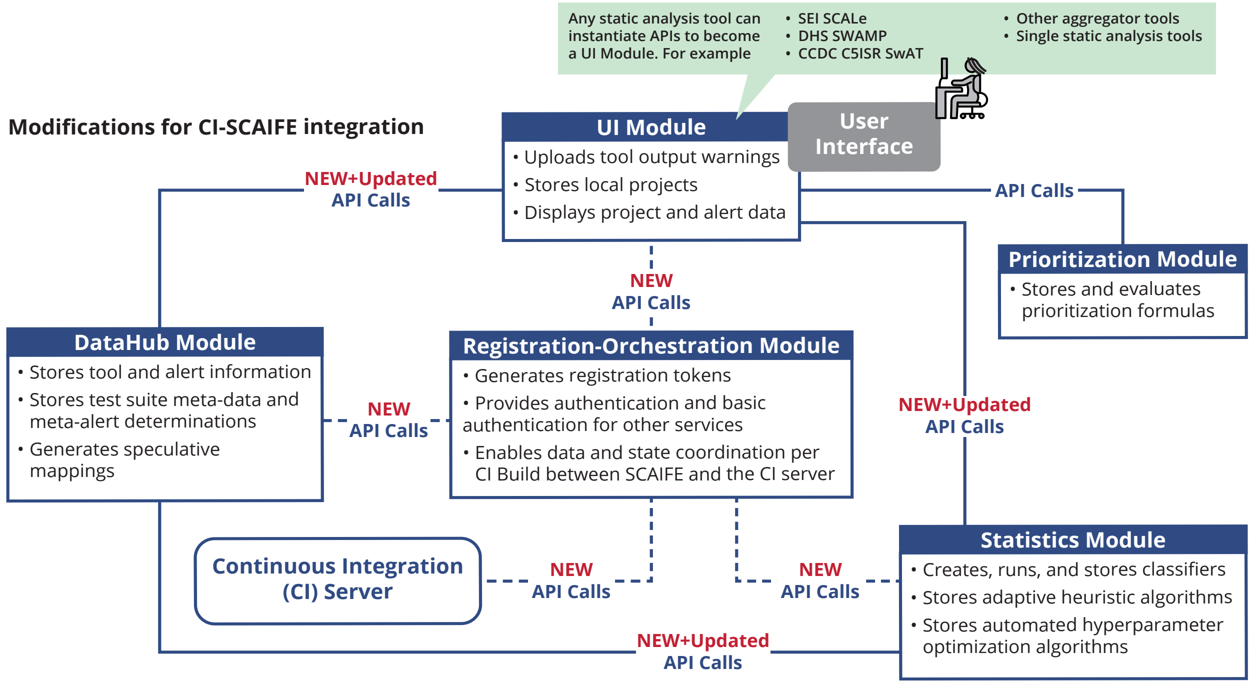
Expected True Positive (e-TP) or
Expected False Positive (e-FP),
and
the rest as Indeterminate (I)





To **overcome barriers to using automated classifiers during CI**, we designed a system **that enables classification to be used in CI builds, including cascading adjudications.**

Modifications for CI-SCAIFE integration



Any static analysis tool can instantiate APIs to become a UI Module. For example
- SEI SCALe
- DHS SWAMP
- CCDC C5ISR SwAT
- Other aggregator tools
- Single static analysis tools

UI Module
- Uploads tool output warnings
- Stores local projects
- Displays project and alert data

Prioritization Module
- Stores and evaluates prioritization formulas

DataHub Module
- Stores tool and alert information
- Stores test suite meta-data and meta-alert determinations
- Generates speculative mappings

Registration-Orchestration Module
- Generates registration tokens
- Provides authentication and basic authentication for other services
- Enables data and state coordination per CI Build between SCAIFE and the CI server

Continuous Integration (CI) Server

Statistics Module
- Creates, runs, and stores classifiers
- Stores adaptive heuristic algorithms
- Stores automated hyperparameter optimization algorithms

FY20 Code and API Artifacts

- (Sep 2020) SCAIFE System v 1.2.2 is released with significant CI-SCAIFE integration progress; it includes five APIs, an HTML manual, SCALe, and the rest of the software system. (collaborators)
- (Sep 2020) SCALe is released for SCALe v. r.6.2.2.2.A. (public)
- (Sep 2020) Five SCAIFE APIs are released. (collaborators, public)
- (Jul 2020) SCAIFE System v 1.1.1 is released with API modules and SCALe automation for CI-SCAIFE integration; the system includes separable SCALe v. r.6.1.1.1.A, five APIs, and an HTML manual. (collaborators)
- (Mar 2020) SCAIFE System v 1.0.0 is released with containers for CI-SCAIFE integration; the system includes a SCALe separable module, new APIs, and an HTML manual. (collaborators)
- (Feb 2020) SCAIFE API v 0.0.9-beta is published. (collaborators, GitHub)
- (Oct 2019) SCAIFE System Beta VM v 2.1 is released with a bill of materials. (collaborators)

FY20 Additional Artifacts

- (Sep 2020) Diff-based cascading experiment artifacts are produced.
- (Sep 2020) A SCAIFE/SCALe HTML manual is released for SCALe v r.7.0.0.0.A. (public, collaborators)
- (Jul 2020) “How to Instantiate SCAIFE API Calls” manual is released. (public)
- (Apr 2020) “Open Dataset RC_Data for Classifier Research” is published. (public)
- (Mar 2020) “How to Test and Review the SCAIFE System v 1.0.0 Release” manual is published. (collaborators)
- (Feb 2020) “SCAIFE API Version 0.0.9-Beta: Reviewer Roadmap” manual is published. (collaborators)

The team developed progressive versions of (1) a design for CI-classifier (CI-SCAIFE) integration and (2) an API definition. The team also implemented a system for modular classification with features to enable CI-integration and to measure performance.

Dr. Lori Flynn
Ebonie McNeil, Matt Sisk, David Svoboda, Hasan Yasar, Joseph Yankel, David Shepard, and Shane Ficorilli